

Article

Eagle-YOLO: Enhancing Real-Time Small Object Detection in UAVs via Multi-Granularity Feature Aggregation

Yan Du ^{1,2}, Zifeng Dai ^{1,2}, Teng Wu ^{1,2}, Quan Zhu ^{1,2}, Changzhen Hu ^{1,2} and Shengjun Wei ^{1,2,*} 

¹ School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100811, China; 3220215217@bit.edu.cn (Y.D.); 3220235383@bit.edu.cn (Z.D.); 3220205106@bit.edu.cn (T.W.); 3220245307@bit.edu.cn (Q.Z.); chzhoo@bit.edu.cn (C.H.)

² Beijing Key Laboratory of Software Security Engineering Technology, Beijing 100811, China

* Correspondence: sjwei@bit.edu.cn

Highlights

What are the main findings?

- The study identifies spectral homogenization as a primary bottleneck in aerial detection. It demonstrates that the proposed Hierarchical Granularity Block (HG-Block) and Cross-Stage Context Modulation (CSCM) effectively preserve fine-grained details while filtering background clutter.
- Extensive experiments on the DUT Anti-UAV and Anti-UAV datasets reveal that Eagle-YOLO achieves a superior speed–accuracy tradeoff, with the lightweight variant surpassing the robust RTMDet-T baseline by 1.67% AP while maintaining a real-time inference speed of 141 FPS.

What are the implications of the main finding?

- The results validate that dynamically aligning receptive fields via Scale-Adaptive Heterogeneous Convolution (SAHC) is critical for distinguishing minute mechanical drones from biological distractors such as birds, thereby challenging the dominance of homogeneous convolutions in real-time detectors.
- The proposed framework offers a practical solution for low-altitude airspace security, proving highly effective for deployment on battery-powered edge monitoring platforms that demand uncompromising precision under strict computational constraints.

Abstract

Real-time object detection in Unmanned Aerial Vehicle (UAV) imagery presents unique challenges, primarily characterized by extreme scale variations and intense background clutter. Existing detectors often suffer from spectral homogenization in which the critical high-frequency details of minute targets are washed out by dominant background signals during feature downsampling. To address this, we propose Eagle-YOLO, a dynamic feature aggregation framework designed to master these complexities without compromising inference speed. We introduce three core innovations: (1) the Hierarchical Granularity Block (HG-Block), which employs a residual granularity injection pathway to function as a detail anchor for tiny objects while simultaneously accumulating semantics for large structures; (2) the Cross-Stage Context Modulation (CSCM) mechanism, which leverages a global context query to filter background redundancy and recalibrate features across network stages; and (3) the Scale-Adaptive Heterogeneous Convolution (SAHC) strategy, which dynamically aligns receptive fields with the inherent scale distribution of aerial data. Extensive experiments on the DUT Anti-UAV dataset demonstrate that Eagle-YOLO achieves a remarkable balance between accuracy and latency. Specifically, our lightweight



Academic Editor: Anastasios Dimou

Received: 12 December 2025

Revised: 25 January 2026

Accepted: 31 January 2026

Published: 3 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Eagle-YOLO-T variant achieves 74.62% AP, surpassing the robust baseline RTMDet-T by 1.67% while maintaining a real-time inference speed of 141 FPS on an NVIDIA RTX 4090 GPU. Furthermore, on the challenging Anti-UAV dataset, our Eagle-YOLOv8-M variant reaches an impressive 94.38% AP_{50}^{val} , outperforming the standard YOLOv8-M by 2.83% and proving its efficacy for edge-deployed aerial surveillance applications.

Keywords: UAV object detection; feature aggregation; real-time detection; multi-granularity

1. Introduction

With the rapid proliferation of consumer-grade drones, the visual detection of unauthorized Unmanned Aerial Vehicles (UAVs) has become a paramount priority for low-altitude airspace security [1]. Unlike general object detection scenarios [2], anti-UAV systems operate in a highly dynamic ground-to-air visual environment. This introduces a unique “small target, vast background” dilemma in which intruder drones often appear as minute clusters of pixels against complex sky domains, fluctuating clouds, or cluttered urban skylines [3]. Furthermore, distinguishing mechanical drones from biological distractors (e.g., flying birds) poses a severe challenge due to their similar scales and appearance. Compounding these difficulties, edge monitoring platforms, which are typically deployed on battery-powered embedded systems, impose rigorous constraints on computational complexity while demanding an uncompromising tradeoff between detection precision and inference latency.

To meet these demands, the community has witnessed the rapid evolution of efficient detectors, exemplified by the YOLO series [4–7]. From the early DarkNet [5] to the recent CSPNet [8] and ELAN [9], these architectures have continuously optimized the speed–accuracy curve. Although these general-purpose detectors achieve impressive performance on natural scene benchmarks, they falter when applied to the anti-UAV domain. Existing solutions often prioritize macro-architectural modifications in the neck to aggregate multi-scale features [10,11]. However, they overlook a critical deficiency inherent in their basic building blocks, namely, the phenomenon of spectral homogenization. Redundancy in standard convolutional blocks causes the network to wash out the faint high-frequency signatures of distant drones, while being overwhelmed by dominant background noise such as cloud edges or tree branches. Although multi-branch structures such as Res2Net [12] attempt to enrich feature diversity, they lack specific mechanisms to filter out such environmental contaminants, leading to high false alarm rates in complex aerial scenes.

In this paper, we fundamentally rethink the design of real-time detectors for the specific task of anti-UAV surveillance. We argue that an effective drone detector must not only be efficient but also granularity-aware and context-sensitive. We identify spectral homogenization as the primary bottleneck, defined as the phenomenon where high-frequency spatial details essential for defining small objects are disproportionately attenuated relative to low-frequency background structures during standard convolutional downsampling. To resolve this, we propose Eagle-YOLO. First, to break the bottleneck of feature homogenization, we redesign the basic building unit into the Hierarchical Granularity Block (HG-Block). By establishing a progressive semantic flow, the HG-Block functions as a “detail anchor” to preserve the pixel-level signatures of tiny targets while accumulating semantics for structure discrimination. Second, to distinguish drones from chaotic environmental distractors, we introduce the Cross-Stage Context Modulation (CSCM) mechanism. This module maintains a global semantic prior to dynamically recalibrate local features, suppressing

spurious activations from birds or clouds. Finally, challenging the dogma of homogeneous convolutions, we formulate the Scale-Adaptive Heterogeneous Convolution (SAHC) strategy. This protocol orchestrates kernel sizes according to feature levels, efficiently aligning the Effective Receptive Field (ERF) with the extreme scale distribution inherent in drone detection tasks.

1.1. Contributions

The main contributions of this paper are summarized as follows:

- We identify spectral homogenization as the primary bottleneck in applying general real-time detectors to UAV imagery. To resolve this, we propose Eagle-YOLO, a novel framework that dynamically orchestrates multi-granular features, thereby establishing a new paradigm for balancing high-precision aerial detection with the strict latency constraints of edge devices.
- We introduce three synergistic components that mimic a raptor's visual system: the HG-Block for extracting progressive fine-grained details, the CSCM for suppressing background clutter via global priors, and the SAHC strategy for aligning receptive fields with extreme scale variations.
- Comprehensive evaluations on authoritative datasets, including DUT Anti-UAV and Anti-UAV, demonstrate the superiority of our method. Eagle-YOLO achieves a remarkable tradeoff between speed and accuracy. For instance, Eagle-YOLOv9-T achieves 79.64% AP_{50}^{val} on the DUT Anti-UAV dataset, outperforming the YOLOv9-T baseline by a significant margin of 5.52% while requiring fewer parameters, thereby validating its practicality for real-world UAV applications.

1.2. Organization

The remainder of this paper is organized as follows: Section 2 reviews the existing literature on real-time object detection and small object detection algorithms tailored for UAVs; Section 3 articulates the proposed Eagle-YOLO framework, detailing the designs of the HG-Block, CSCM mechanism, and SAHC strategy; Section 4 presents the experimental setup, implementation details, and a comprehensive analysis of the results, including ablation studies and comparisons with state-of-the-art (SOTA) methods on the DUT Anti-UAV and Anti-UAV datasets; finally, Section 5 concludes the paper and discusses potential avenues for future research.

2. Related Work

2.1. Real-Time Object Detection

The evolution of object detection has progressed from computationally intensive two-stage frameworks to efficient single-stage architectures. Early two-stage algorithms such as R-CNN [13], Fast R-CNN [14], and Faster R-CNN [15] established the foundation by separating region proposal and classification. Subsequent variants like Mask R-CNN [16] added segmentation capabilities, while Cascade R-CNN [17] improved localization through multi-stage refinement. However, inference speeds often fell short for real-time applications. To address this, single-stage detectors like SSD [18] enhanced by architectures such as DenseNet [19] and the YOLO series were developed to unify localization and classification into a single regression problem.

Since Redmon et al. introduced YOLO [4] and its improved version YOLO9000 [20], the series has dominated real-time detection. YOLOv3 [5] introduced multi-scale predictions using FPN, and YOLOv4 [6] integrated CSPDarknet and Ciou loss. Recent iterations have focused on architectural efficiency and gradient optimization: YOLOX [21] adopted anchor-free heads; YOLOv6 [22] and YOLOv7 [9] optimized re-parameterization and label

assignment; and YOLOR [23] introduced unified implicit knowledge. The widely used YOLOv8 [24] further refined the C2f module for gradient flow.

Most recently, the field has seen rapid advancements aimed at pushing the limits of accuracy and efficiency. YOLOv9 [7] introduced Programmable Gradient Information (PGI) and GELAN to mitigate deep-layer information loss. Concurrently, YOLOv10 [25] explored end-to-end detection without NMS, YOLOv11 [26] enhanced architectural adaptation, and YOLOv12 [27] focused on attention-centric designs. Additionally, transformer-based real-time detectors like RT-DETRv2 [28] and evolved versions such as PP-YOLOE [29] continue to challenge the efficiency boundaries established by purely CNN-based models.

2.2. Small Object Detection

The detection of small targets within UAV imagery presents persistent challenges, primarily driven by the low signal-to-noise ratio inherent in limited-resolution targets and the intense interference from complex unstructured backgrounds [30]. Unlike standard object detection, the visual signatures of drone targets are easily overwhelmed by environmental noise. To address these issues, researchers have extensively explored specialized improvements to general detection frameworks, predominantly focusing on enhancing feature representation through multi-scale fusion, integrating attention mechanisms to focus on salient regions, and employing advanced context modeling to recover lost semantic details [31–33].

Several methods specifically target the feature loss of small objects. ESOD-YOLO [34] employs re-parameterized inverse blocks and waveform FPNs to preserve multi-scale details. RTSOD-YOLO [35] utilizes adaptive spatial attention and triple feature encoding to enhance small-scale inputs, while RE-YOLO [36] introduces spatial extraction attention in the backbone to capture representative semantics. Gold-YOLO [37] proposes a gather-and-distribute mechanism to effectively fuse information across levels. HIC-YOLOv5 [38] and FFCAYOLO [39] introduce specialized heads and feature alignment. TPH-YOLOv5 [40] integrates transformer prediction heads to handle density, while FE-YOLOv5 [41] and CA-YOLO [42] focus on feature enhancement and context-aware modeling respectively. To balance efficiency on edge devices, EAL-YOLO [43] and Edgs-yolov8 [44] incorporate lightweight attention and optimized backbones specifically for drone platforms.

Beyond these architectural modifications, recent advancements have explicitly addressed the domain gap in aerial small object detection through adapter-based mechanisms and specialized training protocols. For instance, AerialFormer [45] introduces a transformer-based adapter to capture long-range dependencies in aerial images, effectively bridging the gap between local CNN features and global context without retraining the entire backbone. Similarly, methods such as Geo-Trax [46] employ curriculum learning and domain-specific training tricks to handle the extreme scale variations and nuisance factors typical of geospatial data. While these approaches improve detection robustness, they often introduce significant computational overhead or require complex training pipelines, which can be limiting for edge deployment scenarios that demand straightforward end-to-end inference.

2.3. UAV Object Detection

The existing methodologies for UAV object detection can be broadly classified based on their inference paradigms and architectural optimizations. Two-stage frameworks, typified by Faster R-CNN [15] and Mask R-CNN [16], prioritize precision by decoupling region proposal from classification. While effective for localized feature extraction, their cascaded nature introduces significant computational latency, rendering them less viable for time-critical aerial missions. To reconcile high-resolution processing with detection speed, inference-level strategies have been introduced. Slicing-Aided Hyper-Inference (SAHI) [47]

adopts a slicing-based approach, performing inference on overlapping high-resolution crops to preserve minute details lost during standard resizing. Similarly, QueryDet [48] utilizes a coarse-to-fine cascade query mechanism to accelerate inference on high-resolution feature maps. However, these methods invariably increase system latency due to multi-pass inference or complex postprocessing, which challenges the strict real-time requirements of onboard monitoring.

Consequently, single-stage detectors, particularly the YOLO lineage, have emerged as the standard for aerial platforms due to their streamlined end-to-end architecture. Early adaptations focused on aligning anchor priors with the statistical distribution of aerial targets. For instance, Hu et al. [49] and Zhang et al. [50] optimized anchor assignments using K-means clustering on multi-scale feature maps, while Zhai et al. [51] and Dadrass Javan et al. [52] enhanced YOLOv4 by restructuring prediction branches to better capture semantic discrepancies across scales. As model architectures evolved, research shifted towards mitigating background noise and feature misalignment. Zhu et al. [40] integrated Transformer Prediction Heads (TPH) and CBAM attention to resolve object occlusion in dense scenes. Similarly, Zhao et al. [53] and Ma et al. [54] embedded global attention mechanisms and normalized Wasserstein distance metrics into the backbone to refine feature discriminability against low-altitude clutter.

Parallel to accuracy improvements, significant efforts have been directed towards lightweighting these models for edge deployment. Strategies range from replacing heavy backbones with compact variants such as MobileNet [55] or the iterative TIBNet [56] to algorithmic compression techniques such as channel pruning [50]. More recent variants have focused on structural efficiency; for example, Wang et al. [57] and Li et al. [58] leveraged depthwise separable convolutions and GhostblockV2 structures to minimize parameter redundancy in YOLOv8-based models, while others such as T-YOLO [59] and VDTNet [60] have introduced dynamic heads and optimized Spatial Pyramid Pooling (SPP) to compensate for the accuracy drop in lightweight networks. Despite these advancements, a critical gap remains in that most methods rely on static kernel designs or heavy attention modules, but struggle to dynamically adapt their receptive fields to the extreme scale variations of drones without incurring excessive computational costs.

3. Methodology

Robust visual detection of unauthorized UAVs remains a paramount challenge in low-altitude airspace security. The task is plagued by the dilemma of small targets against a vast background, in which drones often appear as minute clusters of pixels against complex sky domains or cluttered urban skylines [10,11]. In this section, we articulate the methodology of Eagle-YOLO, a real-time detector engineered to master these challenges. Unlike previous methods that rely on static building blocks, we propose a dynamic feature encoding framework centered on three novel components. First, to capture the diverse spatial details of airborne targets, we design the HG-Block. This structure replaces conventional bottlenecks with a multi-branch topology that enriches inter-channel feature diversity, ensuring robust encoding for both distant point-like drones and proximate targets with distinct mechanical structures. Second, to counteract severe environmental distractors, we introduce the CSCM mechanism. This module leverages a global context query to bridge different network stages, suppressing harmful background redundancy while guiding the focus toward salient airborne threats. Finally, we formulate the SAHC strategy. By strategically orchestrating convolutions with varying kernel sizes, SAHC aligns the ERF with the inherent scale distribution of drone targets, achieving an optimal tradeoff between localization precision and inference speed.

3.1. Overall Architecture of Eagle-YOLO

The architectural framework of Eagle-YOLO is designed to balance the strict computational constraints of edge monitoring devices with the rigorous demands for anti-UAV precision. We select RTMDet [61] as our baseline due to its efficient architecture, which features a CSPDarkNet backbone and CSP-PAFPN neck enhanced by large-kernel (5×5) depth-wise convolutions. While RTMDet effectively expands the receptive field for general object detection, its uniform kernel size and standard bottleneck design struggle to handle the “small target, vast background” dilemma inherent in UAV imagery.

To bridge this gap, Eagle-YOLO diverges from the standard RTMDet architecture in three critical aspects. First, we replace the homogeneous large-kernel blocks with the HG-Block, establishing a multi-branch topology that acts as a detail anchor for tiny objects while aggregating context for larger ones. Second, distinct from the local processing in RTMDet, we introduce the CSCM mechanism to leverage global semantic priors for suppressing environmental distractors. Finally, instead of the fixed 5×5 kernels used in RTMDet, we implement the SAHC strategy (detailed in Section 3.4). This strategy dynamically orchestrates kernel sizes (ranging from 3×3 to 9×9) across stages, ensuring that fine-grained signatures of distant drones are preserved strictly by smaller kernels while broad semantic contexts of proximate targets are captured by larger kernels. To consolidate global context, a Spatial Pyramid Pooling (SPP) module [62] is retained at the end of the backbone.

For the neck, we deploy a Path Aggregation Feature Pyramid Network (PAFPN) [10,11] to effectively fuse multi-level features. Crucially, the standard bottlenecks in the neck are upgraded to our novel HG-Blocks (detailed in Section 3.2), with the SAHC strategy consistently applied across both the neck and the detection head to maximize scale adaptability. Furthermore, to optimize inference latency for real-time monitoring, we adjust the channel depth of the backbone features and introduce three scalable variants: Eagle-YOLO-T, S, and M.

3.2. Granularity-Aware Feature Modeling

Visual perception of UAVs faces a fundamental contradiction: the feature extractor must safeguard high-frequency transients for detecting distant point-like drones while simultaneously abstracting low-frequency semantics to differentiate proximate drones from avian distractors. Standard blocks such as CSP [8] apply uniform filtering across all channels. This often results in feature erosion, where the faint signatures of micro-UAVs are overwhelmed by the high-amplitude gradients of the sky background. To mitigate this, we propose the HG-Block, which operates as a progressive multi-spectral distiller.

The HG-Block diverges from the holistic processing paradigm of previous YOLO architectures by adopting a split-and-aggregate strategy. As illustrated in Figure 1, the input feature tensor $\mathbf{X}_{in} \in \mathbb{R}^{H \times W \times C}$ is projected and partitioned along the channel dimension into M distinct granularity fragments, denoted as $[\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M]$, where each fragment $\mathbf{X}_k \in \mathbb{R}^{H \times W \times \frac{C}{M}}$.

The mechanism utilizes a Cascaded Context Injection (CCI) pathway to enforce granular interaction. Instead of standard residual addition between layers, this pathway establishes a recursive dependency between channel fragments within the same layer. Specifically, the refined output of the preceding branch $\mathbf{H}_{k-1}$ is injected as a semantic prior into the current fragment $\mathbf{X}_k$. This forces the subsequent branches to encode the discrepancy between local texture and global structure. The recursive aggregation is mathematically defined as

$$\mathbf{H}_k = \begin{cases} \mathbf{X}_k, & k = 1 \\ \mathcal{T}_{IBM}(\mathbf{X}_k + \mathbf{H}_{k-1}), & 1 < k \leq M, \end{cases} \quad (1)$$

where $\mathcal{T}_{IBM}(\cdot)$ represents the transformation function of the Inverted Bottleneck Module and $+$ denotes pixel-wise summation. In this hierarchy, the initial branch $\mathbf{H}_1$ serves as a high-frequency preserver, bypassing heavy convolution to strictly retain the pixel-level details essential for micro-target localization. Conversely, deeper branches where $k > 1$ function as context aggregators, progressively widening the effective receptive field. The final output is reconstructed via concatenation: $\mathbf{X}_{out} = \text{Concat}(\mathbf{H}_1, \dots, \mathbf{H}_M)$. By implementing $\mathcal{T}_{IBM}$ with depthwise separable convolutions, the HG-Block achieves granular feature isolation with minimal computational overhead, making it optimal for edge-based anti-drone deployment.

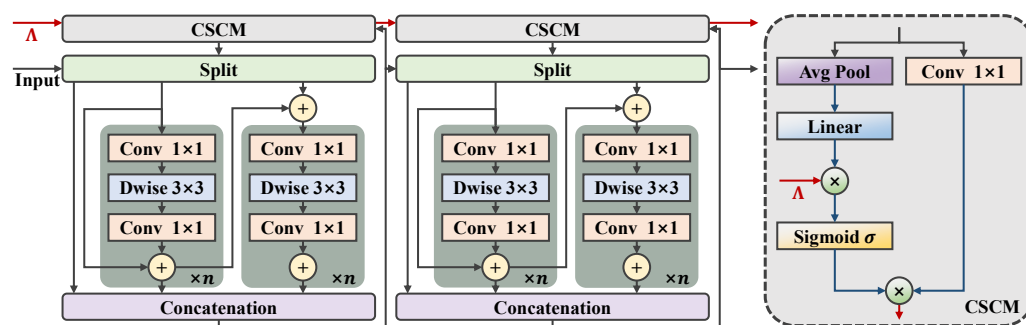


Figure 1. Architecture of the proposed HG-Block. The left part illustrates the recursive split-and-aggregate topology designed to function as a detail anchor for tiny objects. The component enclosed in the dashed box details the CSCM mechanism. Specifically, CSCM utilizes global average pooling to generate a scene descriptor which interacts with learnable global prototypes (Λ), producing dynamic channel-wise weights that effectively filter environmental background clutter before feature aggregation.

It is pertinent to distinguish the proposed HG-Block from the established Res2Net architecture. While both designs utilize a split-and-process strategy to enhance multi-scale representation, their core mechanisms and objectives differ fundamentally. Res2Net focuses on expanding the receptive field by hierarchically adding feature maps from previous splits. In contrast, our HG-Block employs a Cascaded Context Injection pathway designed for granularity preservation. Instead of simple addition, we enforce a recursive dependency where high-frequency residuals from shallow branches are injected into deeper semantic branches. This ensures that the fine-grained texture details of small drones are not lost during semantic abstraction but are instead carried forward as a detail anchor.

3.3. Global Context Aggregation via Query Learning

Despite the granular precision of the HG-Block, its receptive field remains topologically confined. In the domain of counter-UAV operations, this spatial isolation introduces susceptibility to high-frequency environmental noise where the detector may conflate the flapping of avian wings or the edges of cumulus clouds with the mechanical signatures of drones. To rectify this, we engineer the CSCM. This module functions not merely as an attention layer but as a global semantic gatekeeper integrated into the neck of the network.

The underlying mechanism assumes the existence of learnable environmental prototypes that dictate feature importance based on the macroscopic scene composition. We define these latent prototypes as a parameter matrix $\Lambda \in \mathbb{R}^{C \times d_{emb}}$, where C represents the number of feature channels and d_{emb} denotes the embedding dimension. The number of rows C is strictly aligned with the channel width of the input feature map to enable element-wise modulation. The embedding dimension d_{emb} is set to $C/4$ to create a compact bottleneck. This design choice is critical as it forces the network to learn robust low-dimensional semantic abstractions rather than overfitting to specific pixel noise in

the training dataset. The matrix Λ is initialized randomly and updated end-to-end via standard backpropagation effectively encoding the global statistics of the training dataset into a persistent memory.

For an incoming feature tensor X_{in} produced by the HG-Block, we first compress its spatial redundancy into a compact scene descriptor s via global average pooling followed by a linear projection ψ . The CSCM then evaluates the compatibility between the current scene descriptor s and the global prototypes Λ . This interaction generates a channel-wise gating vector which dynamically suppresses background-dominant channels towards zero while amplifying target-relevant signals towards one. The modulation process is mathematically formalized as follows:

$$s = \psi \left(\frac{1}{H \times W} \sum_{h,w} X_{in}^{(h,w)} \right) \quad (2)$$

$$X_{out} = \sigma(\Lambda \cdot s^T) \otimes X_{in} \quad (3)$$

where ψ represents a linear transformation layer mapping the channel dimension from C to d_{emb} , σ denotes the Sigmoid activation function, and $\otimes$ indicates channel-wise broadcasting multiplication. Through this global-to-local interaction, the network learns to implicitly categorize the aerial background style and adjust the feature response gain accordingly. This ensures robust target lock-on even in clutter-rich airspaces without requiring explicit supervision for background types.

3.4. Scale-Adaptive Heterogeneous Kernel Selection

In addition to the micro-design of building blocks, we optimize the macro-architectural choice of convolution kernels. A prevailing limitation in real-time detectors is the exclusive use of homogeneous 3×3 convolutions. While computationally uniform, this one-size-fits-all approach is suboptimal for drone detection which demands a variable visual span.

In a feature pyramid, shallow stages process high-resolution maps rich in fine-grained semantics which are vital for point-like distant drones. Conversely, deep stages handle low-resolution maps. Applying uniform small kernels limits the ERF in deep layers and potentially fragments the semantic integrity of nearby large drones. Conversely, indiscriminately applying large kernels in shallow layers introduces contaminative noise from the sky or ground boundary.

To resolve this, we introduce the SAHC strategy. Instead of relying on runtime kernel prediction, we define a deterministic layer-wise heterogeneity where the kernel size k is strictly coupled with the feature stage s . We formulate a stage-wise kernel function, denoted as $\mathcal{K}(s) = 2s + 1$, to enforce a structural hierarchy.

- Shallow Stages ($s = 1, k = 3$): We utilize compact 3×3 kernels to act as a foveal focus, strictly preserving local pixels for tiny object regression.
- Deep Stages ($s = 2, 3, 4 \rightarrow k = 5, 7, 9$): We progressively expand kernels. These act as peripheral scanners that dramatically enlarge the ERF to capture the holistic structure of proximate targets and distinguish them from similar-sized birds based on structural context.

The specific architectural configuration of this strategy is detailed in Table 1. SAHC constitutes a fixed architectural prior, ensuring deterministic inference speed. To mitigate the computational overhead typically associated with the large kernels shown in the table (e.g., $k = 5, 7, 9$), we employ two strategic optimizations. First, all large-kernel operations are implemented as depth-wise separable convolutions, which significantly reduces the parameter count and floating-point operations compared to standard convolutions. Second,

the kernel size expansion is inversely coupled with the feature map resolution. As observed in Table 1, the largest 9×9 kernels are exclusively applied to Stage-4, where the spatial resolution is downsampled to $1/32$ of the input. Consequently, despite the increased kernel footprint, the actual number of multiplication operations remains manageable. This design ensures that the expanded receptive field achieves a robust tradeoff between structural perception and real-time inference speed.

Table 1. Detailed specifications of the SAHC strategy applied in the backbone.

Block	Layer Type	k	s	p
Downsample-1	Conv-BN-SiLU	3	2	1
Stage-1	HG-Block $\times N_1$ (Foveal Focus)	3	1	1
Downsample-2	Conv-BN-SiLU	3	2	1
Stage-2	HG-Block $\times N_2$ (Local Context)	5	1	2
Downsample-3	Conv-BN-SiLU	3	2	1
Stage-3	HG-Block $\times N_3$ (Semantic Part)	7	1	3
Downsample-4	Conv-BN-SiLU	3	2	1
Stage-4	HG-Block $\times N_4$ (Global Context)	9	1	4

4. Experiments

4.1. Experiment Setup

4.1.1. Implementation Details

Our implementation is constructed upon PyTorch. All experiments are conducted on a high-performance computing cluster equipped with NVIDIA GeForce RTX 4090 GPUs. To ensure training stability and impartiality, we standardize the batch size across devices. However, for the Eagle-YOLO-M variant, the batch size is adjusted accordingly due to the increased memory overhead introduced by the large-kernel SAHC strategy. We strictly standardized the evaluation protocols to ensure a rigorous comparison. For the anti-UAV tasks, we focused exclusively on object detection metrics and did not employ segmentation heads. All models including baselines and our variants were trained and tested with a fixed input resolution of 640×640 pixels. We adopted the standard COCO evaluation metrics, where AP refers to the mean average precision over IoU thresholds from 0.5 to 0.95. The training schedule was fixed at 300 epochs for all lightweight models to ensure full convergence.

In order to adapt to the unique characteristics of the anti-UAV domain, all variants of Eagle-YOLO were trained from scratch. We deliberately avoided initializing with weights pretrained on generic datasets in order to eliminate domain shifts and force the network to learn distinguishing features of airborne targets against complex sky and ground backgrounds from the start.

During the optimization phase, we employed the AdamW optimizer with specific momentum and with weight decay settings where weight decay was disabled for bias and normalization parameters. The learning rate schedule utilized a flat cosine annealing strategy which incorporated a warm-up phase to prevent divergence. To stabilize the gradient updates for detecting erratic UAV motions, we incorporated the Exponential Moving Average (EMA). For the objective functions, we utilized the focal loss [63] to address the extreme class imbalance caused by the vast background versus minute drone targets along with the DIOU loss [64] for precise bounding box regression. Label assignment was governed by the dynamic SimOTA strategy, which is particularly effective for our task since it prioritizes high-quality matches for tiny and distant objects.

To robustly handle challenges such as scale variations and visual similarities between drones and birds, we implemented a comprehensive data augmentation pipeline. Specifically, we applied Mosaic with a probability of 1.0 and MixUp with a probability of 0.15 during the initial 280 training epochs. These heavy augmentations were deactivated in the final 20 epochs in favor of Large Scale Jittering (LSJ) to mitigate the distribution shift between training and testing. For postprocessing, we aligned our protocol with standard practices. The confidence threshold for filtering candidate boxes was set to 0.001 and the IoU threshold for Non-Maximum Suppression (NMS) was fixed at 0.65. Unless otherwise specified, we retained the top 300 high-confidence detections per image for evaluation. Crucially, these configurations were applied uniformly across all baselines and our proposed models to ensure a fair architectural comparison. All specific hyperparameter values are summarized in Table 2.

Table 2. Detailed implementation settings and hyperparameters for Eagle-YOLO training and inference.

Category	Parameter	Value
Environment	GPU Model	NVIDIA RTX 4090
	Framework	PyTorch 1.10
Optimization	Optimizer	AdamW
	Base Learning Rate	1×10^{-4}
	Weight Decay	0.05
	Momentum	0.9
	Batch Size (XS/S)	32 per GPU
	Batch Size (M)	16 per GPU
	Total Epochs	300
	EMA Decay	0.9998
Scheduling	Warm-up Iterations	1000
	Warm-up Initial Factor	1×10^{-5}
	LR Schedule	Flat Cosine Annealing
Augmentation	Mosaic & MixUp	Epochs 1–280
	LSJ (Fine-tuning)	Epochs 281–300
	Input Resolution	640×640
Inference	Confidence Threshold	0.001
	Top- <i>k</i> Detections	300

4.1.2. Datasets

- The **DUT Anti-UAV** [65] dataset proposed in this study consists of two subsets: a detection subset with 10,000 images, which are split into training, validation, and testing sets, containing a total of 10,109 annotated UAV objects; and a tracking subset with 20 video sequences, each averaging 1240 frames. The dataset exhibits several notable characteristics, including a wide range of image resolutions from 160×240 to 3744×5616 , over 35 different UAV models, and diverse outdoor environments such as sky, buildings, and farmland. Importantly, the target objects are relatively small, averaging around 1.3% of the entire image area, with the smallest being only 0.00019%. The target locations are predominantly concentrated in the center of the image but also have a certain horizontal and vertical distribution. The authors used this dataset to evaluate the performance of fourteen detection algorithms and eight tracking algorithms then proposed a novel method that fuses detection and tracking to significantly improve the tracking performance, providing a valuable benchmark for UAV detection and tracking research.
- The **Anti-UAV** [66] dataset is a large-scale multi-modal benchmark specifically designed for UAV tracking. It comprises 318 RGB-T video pairs, including both visible

(RGB) and thermal infrared (IR) sequences captured in diverse environments such as day and night with varying backgrounds such as buildings, clouds, and trees. The dataset features over 580,000 manually annotated bounding boxes, providing rich information for robust tracking evaluation. A significant challenge of Anti-UAV is that the RGB and IR video pairs are unaligned, reflecting real-world complexities. The dataset is divided into training, validation, and test sets, with the test set containing more complex scenarios to rigorously assess tracker performance. It covers a wide range of UAV types and flight patterns, making it a comprehensive platform for developing and testing UAV tracking algorithms, particularly those leveraging multi-modal data fusion.

4.2. Analysis of CSCM

To validate the effectiveness of our CSCM mechanism, particularly in suppressing environmental distractors and highlighting aerial targets, we conducted a series of quantitative and qualitative analyses. These are visualized in Figures 2 and 3.

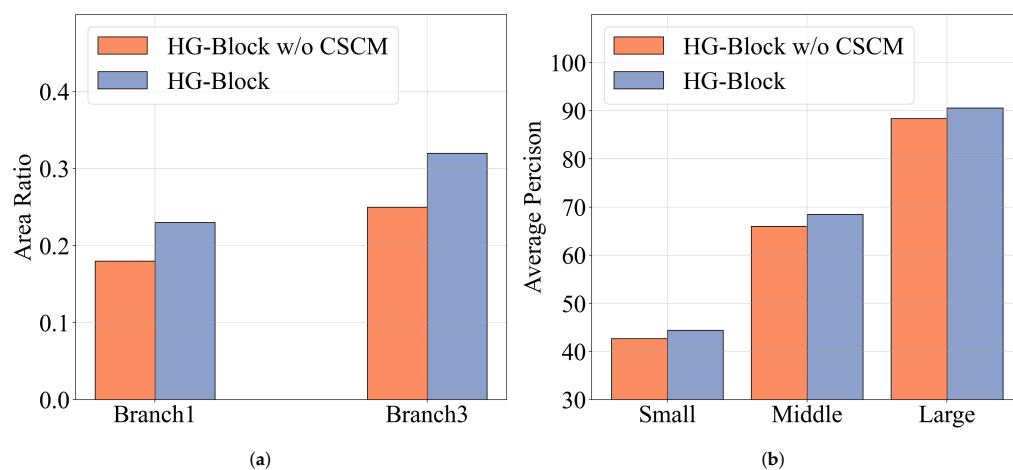


Figure 2. (a) The area ratio of high activation features (value > 0.5) within the GT bounding boxes and (b) the AP across different object scales.

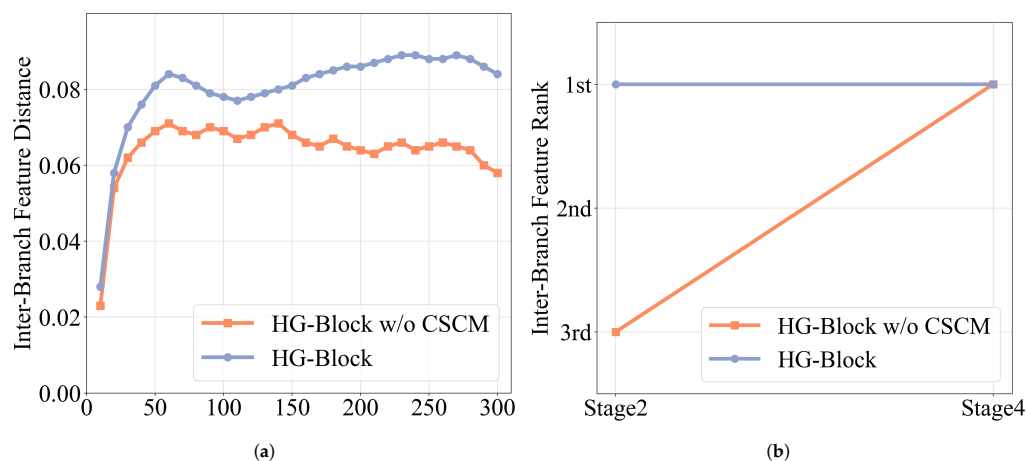


Figure 3. (a) Training dynamics of the inter-branch feature distance and (b) ranking of the average activation value for the target branch.

4.2.1. Activation Coverage

In Figure 2a, we compute the area ratio of high activation features (value > 0.5) within the Ground Truth (GT) bounding boxes. We compare the branch with the smallest receptive field (Branch 1) for minute distant drones and the branch with the largest receptive field

(Branch 3) for proximate structured targets. The results show that the model with CSCM consistently achieves a higher activation ratio compared to the baseline. For small targets specifically, the activation area increases significantly. This implies that without CSCM, the feature map activations tend to drift towards high-frequency background noise (e.g., cloud edges); however, the global query in CSCM acts as a semantic filter, effectively locking the activation strictly within the drone's pixel cluster.

4.2.2. Multi-Scale Performance

Figure 2b presents the Average Precision (AP) across different object scales. It is intuitively evident that CSCM brings a comprehensive improvement. Most notably, the performance on objects in the "Small" category sees a robust boost. This confirms that by modulating local features with global context, the detector can recover the faint signals of distant intruders that would otherwise be overwhelmed by the vast sky background.

4.2.3. Dynamic Weight Evolution

In Figure 3a, we visualize the training dynamics of the inter-branch feature distance (L1 distance) and the attention weight assigned by CSCM. The blue line (with CSCM) shows a rising trend in feature distance compared to the stagnant red line (w/o CSCM). This indicates that CSCM encourages the branches within the HG-Block to learn more distinct and diverse representations that separate drone textures from background textures. Crucially, the yellow line representing the attention weight of the target branch closely follows the trend of the feature distance. This demonstrates that as a branch becomes more discriminative (higher distance), CSCM adaptively assigns it a higher weight, thereby amplifying the signal of the drone against the noise.

4.2.4. Adaptive Branch Ranking

Figure 3b reveals how CSCM rearranges the importance of branches across stages. We compare the ranking of the average activation value for the target branch. In Stage 2 (vital for small object detection), the baseline model ranks the small-receptive-field branch as third, meaning that the network neglects fine-grained details in favor of background features. In contrast, with CSCM this branch leaps to first place. This proves that CSCM successfully forces the shallow layers to prioritize local pixel anomalies (a prerequisite for detecting point-like unauthorized UAVs) while retaining the dominance of large-kernel branches in deeper stages (Stage 4) for holistic structural understanding.

4.3. Analysis of SAHC

Previous studies [67,68] have established the ERF as a critical metric for interpreting the visual span of CNNs. In the specific context of anti-UAV surveillance, the ERF serves as a proxy for the network's "attention scope", determining whether it focuses strictly on the minute drone pixels or is distracted by the surrounding sky domain. In this subsection, we formulate a stage-wise ERF quantification method to investigate the effectiveness of our SAHC.

Unlike static measurements, we adopt a gradient-based back-projection approach to visualize how information flows through the SAHC layers. Let $\mathbf{X}_{in} \in \mathbb{R}^{H \times W \times C}$ denote the input aerial image and $\mathbf{F}^{(s)} \in \mathbb{R}^{H_s \times W_s \times C_s}$ represent the feature tensor generated at the s -th stage of the encoder. To measure the spatial influence, we compute the aggregated gradient response of the central feature vector $\mathbf{f}_{center}^{(s)}$ with respect to the input spatial grid. We define the pixel sensitivity matrix, denoted as $\mathcal{G}^{(s)}$, as follows:

$$\mathcal{G}_{i,j}^{(s)} = \sum_{k=1}^{C_s} \left\| \frac{\partial \mathbf{F}_{u_0, v_0, k}^{(s)}}{\partial \mathbf{X}_{in}^{(i,j)}} \right\|_2 \quad (4)$$

where (u_0, v_0) represents the spatial center of the feature map and (i, j) denotes the coordinates in the input space. Here, $\mathcal{G}^{(s)}$ essentially quantifies how much the (i, j) -th pixel in the sky background contributes to the decision made by the central neuron in stage s .

To visualize this contribution as a heatmap and compress the dynamic range of gradients, we apply a logarithmic projection to obtain the ERF Intensity Map $\mathcal{M}^{(s)}$:

$$\mathcal{M}^{(s)} = \log_{10}(\mathcal{G}^{(s)} + 1). \quad (5)$$

Finally, to quantitatively evaluate the compactness vs. expansiveness of the visual span, we define the effective coverage ratio R_{erf} . Instead of using an absolute threshold, we utilize a relative activation threshold $\tau \in [0, 1]$ to filter the high-response regions:

$$R_{erf}(s, \tau) = \frac{1}{H \times W} \sum_{i,j} \mathbb{I} \left[\frac{\mathcal{M}_{i,j}^{(s)}}{\max(\mathcal{M}^{(s)})} > \tau \right] \quad (6)$$

where $\mathbb{I}[\cdot]$ is the indicator function. A smaller R_{erf} indicates a focused foveal attention suitable for point targets, while a larger R_{erf} indicates a peripheral global attention suitable for structural understanding.

The visual and quantitative analysis based on these metrics is presented in Figure 4. We denote the kernel configuration as $[k_1, k_2, k_3, k_4]$, with our SAHC strategy corresponding to $[3, 5, 7, 9]$.

As observed in Figure 4a, the ERF distribution of Eagle-YOLO exhibits a distinct duality between the fovea and the periphery:

- **In shallow stages ($s = 1, 2$),** the R_{erf} remains extremely compact. This proves that the small kernels in SAHC act as a “detail anchor”, strictly limiting the gradient flow to the drone’s pixel cluster. This prevents the “washing out” of distant targets by prohibiting high-frequency background noise (e.g., cloud textures) from entering the receptive field.
- **In deep stages ($s = 3, 4$):** The R_{erf} expands significantly, surpassing homogeneous settings. This expanded scope allows the network to aggregate context from the entire rotor–fuselage structure. This holistic perception is vital for distinguishing mechanical drones from biological distractors (birds) based on structural integrity rather than local texture.

The quantitative curve in Figure 4b further validates this design. The SAHC curve starts low (minimizing noise) and rises sharply (maximizing context), demonstrating that our strategy effectively aligns the network’s visual span with the inherent multi-scale nature of the anti-UAV task.

4.4. Ablation Study

4.4.1. Ablation Tests on the Proposed Methods

To investigate the individual and combined impact of our proposed innovations, we conducted an ablation study by gradually integrating components from the baseline RTMDet [61] into our final Eagle-YOLO architecture. The quantitative results are reported in Table 3.

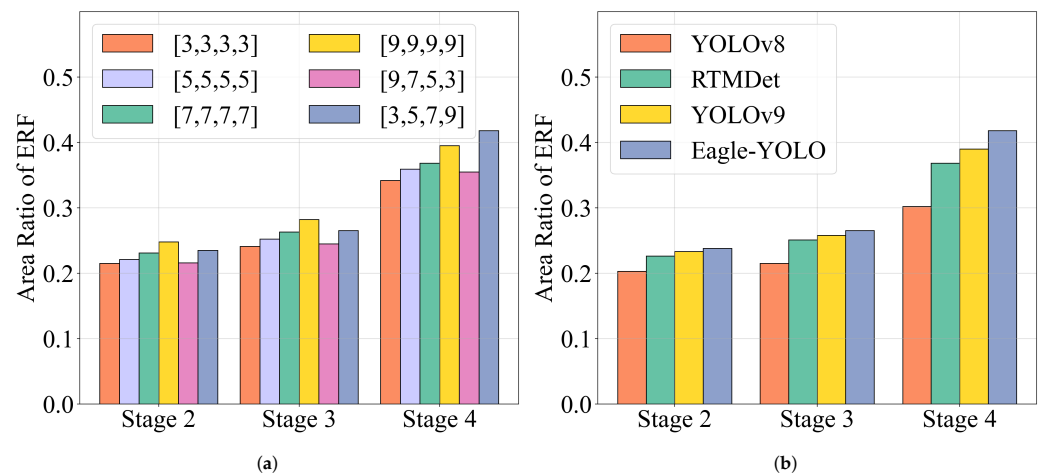


Figure 4. (a) Comparison of the area ratio of ERF under different kernel size settings and (b) comparison of the area ratio of ERF under different real-time detectors.

Table 3. Ablation study on the proposed methods.

Proposed Modules			Accuracy Metrics				Efficiency		
HG-Block	CSCM	SAHC	AP	AP _S	AP _M	AP _L	FPS	#Param. (M)	FLOPs (G)
			72.94	51.67	77.92	90.81	135	4.9	16.2
✓			73.02	52.31	78.02	89.14	165	1.8	8.1
✓	✓		73.12	52.89	78.14	89.52	160	1.8	8.2
✓	✓	✓	74.62	54.39	78.67	92.19	141	2.0	8.4

The first two rows reveal the critical role of the HG-Block in maintaining performance under strict constraints. While the absolute AP improvement of 0.08% from the baseline RTMDet-T appears marginal, it must be interpreted in the context of model complexity. By incorporating the HG-Block, we were able to strategically reduce the channel width of the backbone, resulting in a drastic reduction of parameters by approximately 63% from 4.9 million to 1.8 million and FLOPs by 50% from 16.2G to 8.1G. The fact that the HG-Block sustains and even slightly surpasses the baseline accuracy despite such a significant reduction in capacity proves its effectiveness as a highly efficient feature extractor.

Regarding the specific latency cost of the global context query in CSCM, a comparison between the second and third rows of Table 3 reveals that introducing the CSCM mechanism increases accuracy by 0.1% AP, while the inference speed decreases slightly from 165 FPS to 160 FPS. This corresponds to a marginal incremental latency of approximately 0.19 ms per image. Given the minimal FLOPs increase of 0.1 G and the negligible memory overhead, where peak inference memory remains steady at approximately 420 MB, the CSCM proves to be a highly cost-effective module for real-time applications.

Finally as shown in the fourth row, applying the SAHC strategy introduces the largest computational cost, which drops the FPS to 141. However, this is still faster than the baseline RTMDet-T at 135 FPS. This demonstrates that our heterogeneous kernel design is more efficient than simply scaling up model width or depth. To further ensure that the reported cumulative improvement of 1.67% AP over the baseline is a genuine architectural gain rather than an artifact of random seed selection or training noise, we conducted a statistical significance test over five independent runs, with the results detailed in Table 4. The baseline model exhibited a mean AP of 72.94% with a standard deviation of 0.21, while Eagle-YOLO-T achieved 74.62% with a standard deviation of 0.18. A Welch's *t*-test yielded a *p*-value of less than 0.001, confirming the statistical significance of the performance gap.

Table 4. Statistical significance test over five independent runs.

Model	Mean AP (%)	Std Dev (σ)	<i>p</i> -Value
RTMDet-T	72.94	0.21	<0.001
Eagle-YOLO-T	74.62	0.18	

4.4.2. Ablation Tests on Branch Number

Our HG-Block partitions and propagates the input tensor through multiple granularity branches (N_b) to separate high-frequency details from low-frequency context. However, increasing the branch number expands the depth of the Inverted Bottleneck Module (IBM) while reducing the channel width per branch. To investigate this tradeoff, we conducted an ablation study on N_b , with the results detailed in Table 5. When $N_b = 2$, the granularity separation is insufficient, leading to suboptimal AP. When $N_b = 4$, although the AP reaches 73.66%, the FPS drops significantly to 138 due to increased computational cost. To achieve the optimal balance between inferring minute pixel clusters and structural targets, we designated $N_b = 3$ as the default setting. This configuration offers the best tradeoff between detection accuracy and inference latency on edge devices.

Table 5. Ablation study on the branch number of HG-Block. Here, N_b refers to the number of granularity branches. The baseline is Eagle-YOLO without the SAHC strategy (using standard 3×3 kernels).

N_b	Accuracy Metrics				Efficiency		
	AP	AP _S	AP _M	AP _L	FPS	#Param. (M)	FLOPs (G)
2	71.84	51.22	77.46	88.61	175	1.5	7.4
3	73.11	52.83	78.10	89.52	160	1.8	8.2
4	73.66	53.43	78.64	90.21	138	2.2	9.5

4.4.3. Ablation Tests on the Spatial Dimension of the Query

Our CSCM mechanism maintains a learnable global query $\mathbf{Q}$ to bridge cross-stage information and adaptively modulate local features. The spatial dimension of this query, denoted as $D_s \times D_s$, determines the resolution of the global semantic prior. We investigated the impact of different query sizes, with the results summarized in Table 6. Intriguingly, simply increasing the query dimension does not linearly improve performance. An overly large dimension introduces redundant background noise into the query, confusing the modulator and dropping AP to 72.95%. Conversely, a too-small dimension fails to capture the spatial distribution of cloud cover; specifically, Eagle-YOLO reaches its peak performance of 73.12% AP when $D_s = 4$. This suggests that a 4×4 abstraction is sufficient to represent the global sky/ground context required to suppress distractors.

Table 6. Ablation study on the global query's spatial dimension. Here, D_s refers to the spatial resolution ($D_s \times D_s$) of the query in the CSCM module. The baseline is Eagle-YOLO without the SAHC strategy (using standard 3×3 kernels).

D_s	Accuracy Metrics				Efficiency		
	AP	AP _S	AP _M	AP _L	FPS	#Param. (M)	FLOPs (G)
2×2	72.83	52.12	77.94	89.21	162	1.8	8.2
3×3	73.02	52.65	78.03	89.47	161	1.8	8.2
4×4	73.12	52.86	78.10	89.52	160	1.8	8.2
5×5	73.08	52.72	78.12	89.48	156	1.9	8.2
6×6	72.95	52.41	78.01	89.34	152	1.9	8.3

4.4.4. Ablation Tests on Different Kernel Settings

We performed a comprehensive quantitative comparison using different convolution kernel configurations to evaluate the effectiveness of our SAHC strategy. We explored homogeneous settings and an inverted hierarchy. The results in Table 7 reveal critical insights:

- Homogeneous Small Kernels [3, 3, 3, 3]: While computationally efficient, this setting restricts the ERF in deep stages. Consequently, the AP_L drops to 89.52%, demonstrating that a limited visual span causally prevents the network from perceiving the holistic structure required to detect larger proximate drones.
- Homogeneous Large Kernels [9, 9, 9, 9]: Although this maximizes the receptive field, it significantly degrades AP_S to 50.14%. This decline proves that applying large kernels to high-resolution shallow features introduces excessive environmental noise, which obscures the faint signatures of tiny targets during feature extraction.
- Inverted Hierarchy [9, 7, 5, 3]: This configuration yields the lowest overall performance. It confirms that deep semantic layers strictly require large receptive fields for classification, which small kernels fail to provide, while early large kernels corrupt spatial details.

Our SAHC strategy with kernel sizes [3, 5, 7, 9] combined with extension to the neck and head stands out as the optimal solution. We further analyzed the computational cost relative to the kernel size configuration. As detailed in Table 7, moving from a homogeneous 3×3 setting to our heterogeneous SAHC strategy increases the parameter count by only 0.2 million and the GFLOPs by 0.2G. This marginal increase is attributed to the fact that computationally expensive large kernels are restricted to low-resolution deep layers. In terms of latency, the frame rate decreases from 165 FPS to 141 FPS. This tradeoff is justified by the significant 1.5% AP improvement. It confirms that our layer-wise heterogeneity successfully balances the expansive visual scope required for anti-UAV tasks with strict real-time constraints.

Table 7. Comparison with different kernel size settings of the SAHC strategy. The baseline is Eagle-YOLO without SAHC. Note that the final row represents the complete Eagle-YOLO architecture.

Kernel Settings [k_1, k_2, k_3, k_4]	Accuracy Metrics				Efficiency	
	AP	AP_S	AP_M	AP_L	#Param. (M)	FLOPs (G)
[3, 3, 3, 3] (Baseline)	73.12	52.84	78.13	89.52	1.8	8.2
[5, 5, 5, 5]	73.47	51.92	78.34	90.12	1.9	8.3
[7, 7, 7, 7]	73.58	51.20	78.42	90.85	1.9	8.5
[9, 9, 9, 9]	73.41	50.14	78.52	91.09	2.0	8.7
[11, 11, 11, 11]	73.22	49.54	78.53	91.21	2.1	9.0
[5, 7, 9, 11]	73.67	51.12	78.44	91.17	2.1	9.0
[3, 7, 11, 15]	73.50	50.81	78.42	91.32	2.2	9.2
[9, 7, 5, 3] (Inverted)	72.82	50.04	77.92	89.81	1.9	8.6
[3, 5, 7, 9] (Backbone SAHC)	73.95	53.50	78.55	91.50	1.9	8.3
+Neck-SAHC	74.28	53.94	78.61	91.86	2.0	8.4
+Neck-SAHC + Head-SAHC	74.62	54.39	78.67	92.19	2.0	8.4

4.4.5. Analysis of Image Resolution

Finally, we investigated the correlation between input image resolution and the robustness of our multi-scale design. During inference, we applied Test Time Augmentation (TTA) with varying scales. The results are provided in Table 8.

Table 8. Comparison of image resolution robustness. We evaluated the performance stability across varying input scales.

Model	Resolution	AP	AP _S	AP _M	AP _L
RTMDet-T	320 × 320	65.82	40.58	71.47	87.66
Eagle-YOLO-T		67.55	43.31	71.68	89.68
RTMDet-T	640 × 640	72.94	51.67	77.95	90.81
Eagle-YOLO-T		74.62	54.39	78.67	92.19
RTMDet-T	1280 × 1280	74.13	59.32	77.51	87.61
Eagle-YOLO-T		75.98	62.06	78.69	88.64
RTMDet-T	TTA	74.85	59.57	79.24	87.61
Eagle-YOLO-T		77.86	62.38	80.37	92.37

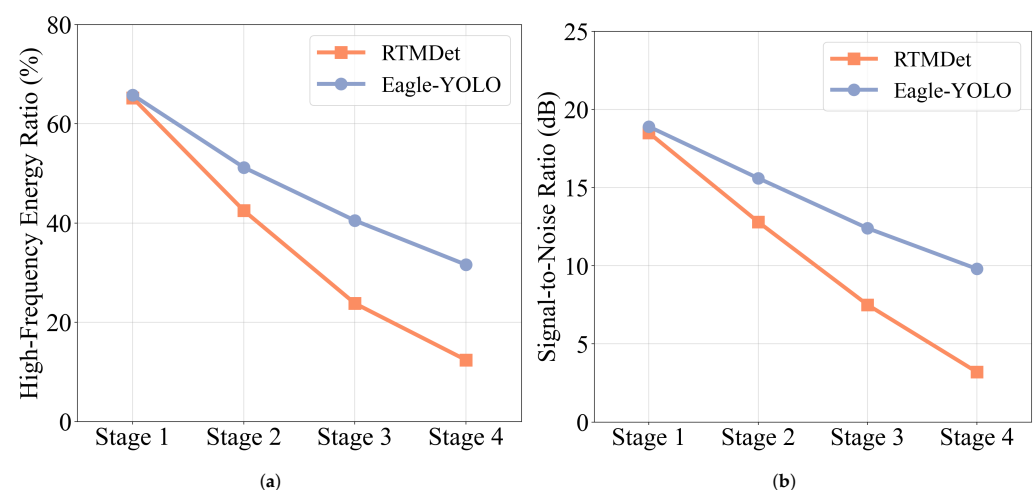
The experimental results demonstrate a consistent trend of improved overall AP as the resolution increases:

- At high resolution (1280), AP_S sees a dramatic boost to 62.06%. This is expected, as higher resolution resolves distant sub-pixel drones into discernible shapes.
- At low resolution (320), AP_S naturally drops due to information loss; however, Eagle-YOLO maintains a surprisingly high AP_L compared to RTMDet-T.

This stability at low resolutions verifies the effectiveness of our SAHC and HG-Block protocols. Even with limited pixel information, the network leverages the expanded ERF and granular feature injection to identify drone structures, ensuring its reliability even when video transmission quality degrades in real-world UAV operations.

4.4.6. Quantitative Analysis of Spectral Homogenization

To empirically verify the alleviation of spectral homogenization, we conducted a frequency domain analysis and signal-to-noise ratio (SNR) monitoring across network stages, as visualized in Figure 5.

**Figure 5.** (a) Comparison of HFER across network stages and (b) evolution of feature SNR.

High-Frequency Preservation: As shown in Figure 5a, the baseline RTMDet exhibits a sharp decline in the High-Frequency Energy Ratio (HFER), dropping drastically from 65.2% in Stage 1 to 12.4% in Stage 4. This confirms our hypothesis that standard strided convolutions act as aggressive low-pass filters, eroding the critical boundary details of small targets. In contrast, Eagle-YOLO maintains a robust 31.6% HFER in the deepest stage. This demonstrates that the granularity injection pathway in our HG-Block effectively

functions as a detail anchor, preventing high-frequency textures from being washed out during downsampling.

SNR Enhancement: Figure 5b further illustrates the evolution of feature discriminability. While the baseline SNR degrades rapidly to 3.2 dB due to the dominance of background clutter in deep layers, Eagle-YOLO maintains a significantly higher SNR of 9.8 dB at Stage 4. This substantial gap indicates that the global semantic priors provided by the CSCM mechanism successfully suppress environmental noise, while the SAHC strategy preserves the target signal intensity.

4.4.7. Sensitivity Analysis of Hyperparameters

To verify that the reported performance gains stem from our architectural innovations rather than overfitting to specific postprocessing configurations, we conducted a sensitivity analysis on key hyperparameters such as the NMS threshold and the number of retained detections (Top-k).

Table 9 presents the performance of Eagle-YOLO-T compared to the RTMDet-T baseline under varying settings. When the NMS IoU threshold varies from 0.50 to 0.70, the absolute AP values fluctuate for both models, as expected; however, Eagle-YOLO-T consistently outperforms the baseline by a margin of 1.3% to 1.7% AP. Similarly, adjusting the Top-k value between 100, 300, and 500 impacts the recall rate, but does not alter the relative superiority of our method; for instance, even with a restricted Top-k of 100, our model maintains a 1.55% lead. This consistent superiority confirms that the improvements are robust and intrinsic to the proposed HG-Block and SAHC strategy rather than being artifacts of specific hyperparameter tuning.

Table 9. Sensitivity analysis of postprocessing hyperparameters on the DUT Anti-UAV dataset.

Parameter	Value	AP (%)		Gap (%)
		RTMDet-T	Eagle-YOLO-T	
NMS Threshold	0.50	72.85	74.48	+1.63
	0.60	73.10	74.72	+1.62
	0.65 (Default)	72.94	74.62	+1.68
	0.70	72.45	73.80	+1.35
Top-k Detections	100	71.80	73.35	+1.55
	300 (Default)	72.94	74.62	+1.68
	500	72.98	74.65	+1.67

4.4.8. Robustness Analysis Against Environmental Distractors

To move beyond qualitative visualizations and provide concrete evidence of disturbance suppression, we conducted a quantitative error analysis based on background complexity. We partitioned the DUT Anti-UAV test set into two distinct subsets. The first subset is labeled Simple Backgrounds, which consists of clear sky scenes with minimal noise. The second subset is labeled Complex Backgrounds, which includes challenging scenarios featuring dense cloud layers, urban buildings, vegetation, and low-altitude clutter where false positives typically occur.

We adopted False Positives Per Image (FPPI) as a key metric to measure the susceptibility of the model to environmental distractors. A false positive was defined as a detection with a confidence score higher than 0.25 that does not overlap with any ground truth drone. Table 10 summarizes the performance of the baseline RTMDet-T and our Eagle-YOLO-T across these scenarios.

On the Simple Backgrounds subset, both models exhibit high performance with comparable AP values, indicating that standard detectors handle clear skies effectively.

However, a significant divergence appears on the Complex Backgrounds subset. The baseline model suffers a performance degradation of 13.4% in AP and exhibits a high FPPI of 0.45, suggesting that it frequently misclassifies cloud edges or bird-like objects as drones. In contrast, Eagle-YOLO-T demonstrates superior robustness, maintaining a significantly higher AP of 68.12% in complex scenes and reducing the FPPI to 0.18. This reduction of approximately 60% in false alarms quantitatively confirms that the global priors from the CSCM mechanism and that the structural filtering from the SAHC strategy effectively suppress non-target gradients, ensuring reliable operation even in clutter-rich environments.

Table 10. Quantitative robustness analysis on the DUT Anti-UAV dataset.

Scenario	Model	Accuracy Metrics		Error Metrics	
		AP ₅₀ (%)	Gap (%)	FPPI	Reduction
Simple Backgrounds	RTMDet-T	84.50	-	0.05	-
	Eagle-YOLO-T	85.10	+0.60	0.04	−20.0%
Complex Backgrounds	RTMDet-T	61.35	-	0.45	-
	Eagle-YOLO-T	68.12	+6.77	0.18	−60.0%

4.5. Comparison with the SOTA Methods

We conducted a comprehensive benchmarking on two authoritative datasets, DUT Anti-UAV and Anti-UAV, to evaluate the tradeoff between speed and accuracy. To demonstrate the scalability and universality of our approach, we developed three variants of Eagle-YOLO targeting distinct computational scales. First, Eagle-YOLOv9-T is built upon the YOLOv9-T baseline to optimize efficiency for edge devices. Second, to verify effectiveness on the latest architectures, we developed Eagle-YOLOv10-S based on YOLOv10-S. Finally, for scenarios demanding high precision, we extended the architecture to Eagle-YOLOv8-M based on the YOLOv8-M baseline. To ensure a fair comparison, we re-implemented and retrained all models from scratch using their official codebases, avoiding any pretrained weights. Furthermore, we report inference speeds on both a server-grade NVIDIA RTX 4090 and an embedded NVIDIA Jetson Orin NX to rigorously assess real-world deployability. The detailed comparative results are presented in Table 11.

- **Performance on DUT Anti-UAV Dataset:** Eagle-YOLO demonstrates a distinct advantage over both general real-time detectors and domain-specific methods in terms of the accuracy–latency tradeoff. In the lightweight category, Eagle-YOLOv9-T achieves an AP₅₀^{val} of 79.64% and exceeds the robust baseline YOLOv9-T by a margin of 5.52%. Crucially, it outperforms the specialized UAV detector EDGS-YOLOv8 (score of 76.85%) while requiring significantly fewer parameters, specifically 2.0 million compared to 4.2 million. In terms of speed, it maintains a remarkable 145 FPS on the RTX 4090 and achieves 82 FPS on the Jetson Orin NX, confirming its suitability for high-speed edge deployment. Within the small model group, Eagle-YOLOv10-S attains an AP₅₀^{val} of 82.37%. Notably, it is the fastest model in this category, reaching 128 FPS on the RTX 4090 and 43 FPS on the Jetson, surpassing both the baseline YOLOv10-S and the transformer-enhanced TPH-YOLOv5. For high-precision scenarios, our large-capacity Eagle-YOLOv8-M reaches a SOTA AP₅₀^{val} of 90.26%. Thanks to the efficient design of our HG-Block, which reduces FLOPs, this variant achieves 92 FPS on the RTX 4090. This is significantly faster than the standard YOLOv8-M at 71 FPS and RTMDet-M at 70 FPS, proving that high accuracy does not necessarily come at the cost of high latency.

- **Performance on Anti-UAV Dataset:** To verify the robustness of our method across differing data distributions, we extended the evaluation to the Anti-UAV dataset. As detailed in Table 11, our method consistently leads the performance rankings. Eagle-YOLOv9-T achieves an AP_{50}^{val} of 85.64% and outperforms TransVisDrone (score of 81.30%) by 4.34% while running significantly faster (141 FPS vs. 105 FPS). Notably, Eagle-YOLOv8-M advances the performance boundary to an AP_{50}^{val} of 94.38% and an $AP_{50:95}^{val}$ of 66.07%. This represents an improvement of 2.83% over the standard YOLOv8-M. These consistent gains across datasets validate that our granularity-aware design successfully mitigates the information loss inherent in standard downsampling and delivers optimal performance for diverse aerial monitoring platforms.

Table 11. Comparison with SOTA methods on the DUT Anti-UAV and Anti-UAV datasets. Note that [†] indicates domain-specific methods tailored for UAV detection.

Model	AP_{50}^{val} (%)		$AP_{50:95}^{val}$ (%)		#Param. (M)	FLOPs (G)	FPS	
	DUT Anti-UAV	Anti-UAV	DUT Anti-UAV	Anti-UAV			4090	Jetson
RTMDet-T [61]	73.53	78.29	47.51	50.84	4.9	16.2	133	52
YOLOv6-T [22]	75.12	80.25	49.21	52.49	9.7	24.8	110	38
YOLOv7-T [9]	73.81	78.43	47.92	51.36	6.2	13.8	146	60
YOLOv9-T [7]	74.12	78.96	48.54	51.92	2.1	8.2	162	92
TIBNet [†] [56]	65.23	69.10	38.45	41.20	1.3	3.5	283	132
VDNet [†] [60]	68.60	72.50	41.20	44.80	3.9	13.5	140	58
EDGS-YOLOv8 [†] [44]	76.85	81.15	48.90	52.40	4.2	10.1	154	76
Eagle-YOLOv9-T	79.64	85.64	49.34	54.65	2.0	8.4	145	82
YOLOv6-S [22]	78.56	83.17	52.88	55.93	17.2	43.8	96	22
YOLOv9-S [7]	78.36	83.45	52.31	55.83	7.1	26.8	121	38
YOLOv10-S [25]	78.94	83.76	52.53	56.17	8.1	24.8	124	41
RTMDet-S [61]	79.15	84.06	53.24	56.71	8.9	29.6	113	35
Gold-YOLO-S [37]	80.55	85.12	54.12	57.43	21.5	46.0	88	20
YOLOv8-S [24]	79.47	84.62	55.67	59.18	11.2	28.8	118	36
TPH-YOLOv5 [†] [40]	81.67	85.15	54.12	58.54	8.5	18.2	92	32
TransVisDrone [†] [69]	80.38	86.21	54.39	58.64	12.4	25.6	105	39
Eagle-YOLOv10-S	82.37	87.37	56.75	61.49	7.1	23.0	128	43
YOLOv10-M [25]	85.13	89.92	58.22	62.05	16.5	63.0	84	18
RT-DETR-R18 [70]	83.54	88.13	56.44	59.87	20.0	60.0	75	16
RTMDet-M [61]	85.52	90.35	58.42	62.19	24.7	78.6	70	15
YOLOv8-M [24]	86.46	91.55	59.82	63.74	25.9	79.2	71	14
RT-DETR-R34 [70]	86.83	91.42	59.13	62.91	31.0	90.2	60	11
YOLOv6-M [22]	84.85	89.24	57.14	60.58	34.3	81.4	65	13
Eagle-YOLOv8-M	90.26	94.38	62.36	66.07	25.9	70.4	92	19

To more intuitively analyze the interpretability of Eagle-YOLO in complex aerial environments, we utilized GradCAM [71] to visualize the class activation maps of our model. As illustrated in Figure 6, the model exhibits a laser-like focus, precisely locking onto minute drone targets regardless of the background complexity. Crucially, the high-activation regions are strictly confined to the drone's signature, demonstrating that the network successfully ignores environmental distractors such as cloud edges or buildings. This visual evidence confirms that our CSCM mechanism and SAHC strategy effectively filter out sky noise and guide the network to concentrate on salient foreground features even in challenging low-contrast scenarios.

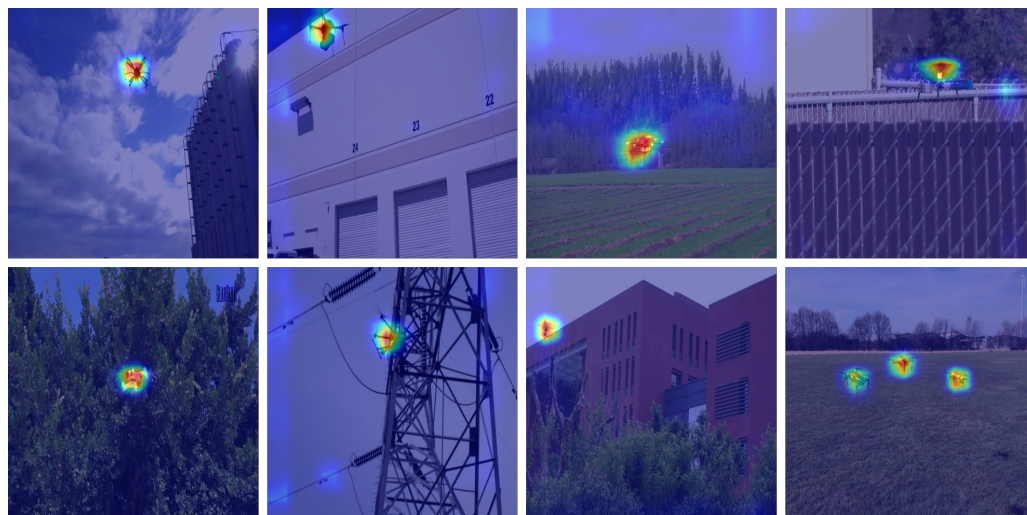


Figure 6. Visualization of class activation maps generated by Eagle-YOLO.

5. Conclusions

In this paper we have presented Eagle-YOLO, a novel real-time object detector specifically engineered to address the “small target, vast background” dilemma in UAV imagery. By rethinking the fundamental building blocks of convolutional networks, we identify feature homogenization as a critical bottleneck and propose a synergistic solution. Our introduced HG-Block successfully preserves fine-grained details for small targets through progressive semantic distillation, boosting the detection of small objects (AP_S) by 2.2% over the baseline. Complementing this, the CSCM mechanism effectively suppresses environmental distractors by integrating global contextual priors, while the SAHC strategy ensures optimal receptive field alignment across varying object scales, leading to a substantial 2.69% increase in large object detection (AP_L). Empirical results on the DUT Anti-UAV dataset confirm that Eagle-YOLO achieves SOTA performance, with our Eagle-YOLOv8-M model reaching 90.26% AP_{50}^{val} , significantly outperforming robust baselines in both accuracy and inference speed. These findings validate Eagle-YOLO as a highly effective and practical solution for real-world resource-constrained aerial surveillance tasks. Future work will explore extending this framework to multi-modal data and further optimizing it for more diverse aerial scenarios.

Author Contributions: Conceptualization, Y.D. and S.W.; methodology, T.W.; software, Y.D.; validation, Y.D., Z.D. and C.H.; formal analysis, Z.D.; investigation, Y.D.; resources, Y.D.; data curation, Q.Z.; writing—original draft preparation, Y.D. and Q.Z.; writing—review and editing, Z.D.; visualization, C.H.; supervision, S.W.; project administration, C.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data supporting the findings of this study are available from author S. Wei upon reasonable request.

Acknowledgments: The authors acknowledge the editors and the anonymous referees for their valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Gupta, L.; Jain, R.; Vaszkun, G. Survey of Important Issues in UAV Communication Networks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1123–1152.
- Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H.Y. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv* **2022**, arXiv:2203.03605.
- Studiawan, H.; Grispos, G.; Choo, K.R. Unmanned Aerial Vehicle (UAV) Forensics: The Good, The Bad, and the Unaddressed. *Comput. Secur.* **2023**, *132*, 103340.
- Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
- Redmon, J.; Farhadi, A. Yolo3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
- Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolo4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
- Wang, C.Y.; Yeh, I.H.; Mark Liao, H.Y. Yolo9: Learning what you want to learn using programmable gradient information. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024; Springer: Berlin/Heidelberg, Germany, 2024; pp. 1–21.
- Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
- Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
- Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
- Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
- Gao, S.H.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662.
- Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
- Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
- Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149.
- He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *arXiv* **2017**, arXiv:1703.06870.
- Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
- Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
- Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
- Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
- Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YoloX: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
- Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
- Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. You only learn one representation: Unified network for multiple tasks. *arXiv* **2021**, arXiv:2105.04206.
- Glenn, J. Yolo8. 2023. Available online: <https://github.com/ultralytics/ultralytics/tree/main> (accessed on 30 January 2026).
- Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. Yolo10: Real-time end-to-end object detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011.
- Khanam, R.; Hussain, M. Yolo11: An overview of the key architectural enhancements. *arXiv* **2024**, arXiv:2410.17725.
- Tian, Y.; Ye, Q.; Doermann, D. Yolo12: Attention-centric real-time object detectors. *arXiv* **2025**, arXiv:2502.12524.
- Lv, W.; Zhao, Y.; Chang, Q.; Huang, K.; Wang, G.; Liu, Y. Rt-detr2: Improved baseline with bag-of-freebies for real-time detection transformer. *arXiv* **2024**, arXiv:2407.17140.

29. Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; Wang, G.; Dang, Q.; Wei, S.; Du, Y.; et al. PP-YOLOE: An evolved version of YOLO. *arXiv* **2022**, arXiv:2203.16250.
30. Sommer, L.; Schumann, A.; Müller, T.; Schuchert, T.; Beyerer, J. Flying object detection for automatic UAV recognition. In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–6.
31. Minaeian, S.; Liu, J.; Son, Y.J. Vision-based target detection and localization via a team of cooperative UAV and UGVs. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *46*, 1005–1016.
32. Bao, Z. The UAV Target Detection Algorithm Based on Improved YOLO V8. In Proceedings of the International Conference on Image Processing, Machine Learning and Pattern Recognition, Guangzhou, China, 13–15 September 2024; pp. 264–269.
33. Verma, T.; Singh, J.; Bhartari, Y.; Jarwal, R.; Singh, S.; Singh, S. SOAR: Advancements in Small Body Object Detection for Aerial Imagery Using State Space Models and Programmable Gradients. *arXiv* **2024**, arXiv:2405.01699.
34. Luo, J.; Liu, Z.; Wang, Y.; Tang, A.; Zuo, H.; Han, P. Efficient Small Object Detection You Only Look Once: A Small Object Detection Algorithm for Aerial Images. *Sensors* **2024**, *24*, 7067.
35. Zhang, S.; Yang, X.; Geng, C.; Li, X. A Reparameterization Feature Redundancy Extract Network for Unmanned Aerial Vehicles Detection. *Remote Sens.* **2024**, *16*, 4226.
36. Liu, B.; Mo, P.; Wang, S.; Cui, Y.; Wu, Z. A Refined and Efficient CNN Algorithm for Remote Sensing Object Detection. *Sensors* **2024**, *24*, 7166.
37. Wang, C.; He, W.; Nie, Y.; Guo, J.; Liu, C.; Wang, Y.; Han, K. Gold-YOLO: Efficient object detector via gather-and-distribute mechanism. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 51094–51112.
38. Tang, S.; Zhang, S.; Fang, Y. HIC-YOLOv5: Improved YOLOv5 for small object detection. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 6614–6619.
39. Zhang, Y.; Ye, M.; Zhu, G.; Liu, Y.; Guo, P.; Yan, J. FFCA-YOLO for small object detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–15.
40. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 2778–2788.
41. Wang, M.; Yang, W.; Wang, L.; Chen, D.; Wei, F.; KeZiErBieKe, H.; Liao, Y. FE-YOLOv5: Feature enhancement network based on YOLOv5 for small object detection. *J. Vis. Commun. Image Represent.* **2023**, *90*, 103752.
42. Shen, L.; Lang, B.; Song, Z. CA-YOLO: Model optimization for remote sensing image object detection. *IEEE Access* **2023**, *11*, 64769–64781.
43. Wang, J.; Sun, Y.; Lin, Y.; Zhang, K. Lightweight substation equipment defect detection algorithm for small targets. *Sensors* **2024**, *24*, 5914.
44. Huang, M.; Mi, W.; Wang, Y. Edgs-yolov8: An improved yolov8 lightweight uav detection model. *Drones* **2024**, *8*, 337.
45. Hanyu, T.; Yamazaki, K.; Tran, M.; McCann, R.A.; Liao, H.; Rainwater, C.; Adkins, M.; Cothren, J.; Le, N. AerialFormer: Multi-resolution transformer for aerial image segmentation. *Remote Sens.* **2024**, *16*, 2930.
46. Fonod, R. Geo-trax: A Comprehensive Framework for Georeferenced Vehicle Trajectory Extraction from Drone Imagery. 2025. Available online: <https://infoscience.epfl.ch/entities/product/cfa285a9-0035-4ca6-a728-831caf2b5bee/datasetdetails> (accessed on 30 January 2026).
47. Akyon, F.C.; Altinuc, S.O.; Temizel, A. Slicing aided hyper inference and fine-tuning for small object detection. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 966–970.
48. Yang, C.; Huang, Z.; Wang, N. QueryDet: Cascaded sparse query for accelerating high-resolution small object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13668–13677.
49. Hu, Y.; Wu, X.; Zheng, G.; Liu, X. Object detection of UAV for anti-UAV based on improved YOLO v3. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8386–8390.
50. Zhang, X.; Fan, K.; Hou, H.; Liu, C. Real-time detection of drones using channel and layer pruning, based on the yolov3-spp3 deep learning algorithm. *Micromachines* **2022**, *13*, 2199.
51. Zhai, H.; Zhang, Y. Target Detection of Low-Altitude UAV Based on Improved YOLOv3 Network. *J. Robot.* **2022**, *2022*, 4065734.
52. Dadras Javan, F.; Samadzadegan, F.; Gholamshahi, M.; Ashatari Mahini, F. A modified YOLOv4 Deep Learning Network for vision-based UAV recognition. *Drones* **2022**, *6*, 160.
53. Zhao, Y.; Ju, Z.; Sun, T.; Dong, F.; Li, J.; Yang, R.; Fu, Q.; Lian, C.; Shan, P. Tgc-yolov5: An enhanced yolov5 drone detection model based on transformer, gam & ca attention mechanism. *Drones* **2023**, *7*, 446.
54. Ma, J.; Huang, S.; Jin, D.; Wang, X.; Li, L.; Guo, Y. LA-YOLO: An effective detection model for multi-UAV under low altitude background. *Meas. Sci. Technol.* **2024**, *35*, 055401.

55. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 1314–1324.
56. Sun, H.; Yang, J.; Shen, J.; Liang, D.; Ning-Zhong, L.; Zhou, H. TIB-Net: Drone detection network with tiny iterative backbone. *IEEE Access* **2020**, *8*, 130697–130707.
57. Wang, C.; Meng, L.; Gao, Q.; Wang, J.; Wang, T.; Liu, X.; Du, F.; Wang, L.; Wang, E. A lightweight UAV swarm detection method integrated attention mechanism. *Drones* **2022**, *7*, 13.
58. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A modified YOLOv8 detection network for UAV aerial image recognition. *Drones* **2023**, *7*, 304.
59. Bai, B.; Wang, J.; Li, J.; Yu, L.; Wen, J.; Han, Y. T-YOLO: A lightweight and efficient detection model for nutrient buds in complex tea-plantation environments. *J. Sci. Food Agric.* **2024**, *104*, 5698–5711.
60. Zhou, X.; Yang, G.; Chen, Y.; Li, L.; Chen, B.M. VDTNet: A high-performance visual network for detecting and tracking of intruding drones. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 9828–9839.
61. Lyu, C.; Zhang, W.; Huang, H.; Zhou, Y.; Wang, Y.; Liu, Y.; Zhang, S.; Chen, K. RtmDET: An empirical study of designing real-time object detectors. *arXiv* **2022**, arXiv:2212.07784.
62. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916.
63. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
64. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
65. Zhao, J.; Zhang, J.; Li, D.; Wang, D. Vision-Based Anti-UAV Detection and Tracking. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25323–25334.
66. Jiang, N.; Wang, K.; Peng, X.; Yu, X.; Wang, Q.; Xing, J.; Li, G.; Zhao, J.; Guo, G.; Han, Z. Anti-UAV: A large multi-modal benchmark for UAV tracking. *arXiv* **2021**, arXiv:2101.08466.
67. Ding, X.; Zhang, X.; Han, J.; Ding, G. Scaling up your kernels to 31x31: Revisiting large kernel design in CNNs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11963–11975.
68. Luo, W.; Li, Y.; Urtasun, R.; Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*.
69. Sangam, T.; Dave, I.R.; Sultani, W.; Shah, M. Transvisdrone: Spatio-temporal transformer for vision-based drone-to-drone detection in aerial videos. *arXiv* **2022**, arXiv:2210.08423.
70. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974.
71. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.